SPRING THE BEST

MSX COMPUTER MAGAZINE

N.13 Sped in abb. post. Gr. III L. 9.000

MSX BANK

Hai programmi originali, esclusivamente pensati e fatti da te? Mandaceli in visione e, se verranno pubblicati, saranno certamente compensati a partire da lire 100 mila in su (a seconda del tipo di programma). Se sei fantasioso, bravo e veloce oltre che ordinato, puoi arrivare a guadagnare un bel gruzzoletto! Naturalmente il tuo nome apparirà stampato sulla rivista come valido collaboratore. Fruga nella tua fantasia e mandaci non solo arcade ma anche utility inedite, giochi di società particolari, interessanti routine in linguaggio macchina ed avventure con grafica (invia sempre anche la soluzione comando per comando in sequenza) allegando anche caratteristiche tecniche dei programmi, listati, stampate di screen etc. Se decidi di inviare un programma (indirizza a MSX Computer Magazine, c.so Vitt. Emanuele 15, 20122 Milano) segui queste regole:

1) salva il programma su entrambi i lati di una cassetta;
2) usa 1 cassetta per ogni programma;
3) scrivi il tuo nome, quello del programma e quello della macchina sulla cassetta;
4) accludi un foglio dattiloscritto dove spiegherai a cosa serve il programma, come si usa, quali tasti usare, lo scopo. Non proteggete i programmi! Dobbiamo guardarli dentro. Non mettete il vostro indirizzo all'interno del programma (basta «by... nome e cognome»).

Rispondiamo sempre a tutti e, soprattutto, manteniamo la parola data. Coraggio dunque, fatevi vivi!

ATTENZIONE: I programmi inviati debbono essere assolutamente inediti, a noi ceduti in esclusiva. L'editore si riserva di rifiutare programmi anche già accettati e di non compensare programmi già pubblicati qualora si verificasse che gli stessi programmi (anche quando con titoli diversi) siano stati ceduti o pubblicati da altri.

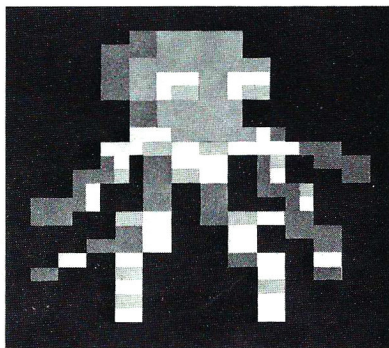
MSX Computer Magazine è edita da Arcadia srl,
C.so Vitt. Emanuele 15, Milano.
Tel. 02/706329 (solo giovedì h. 15-18).
Una copia L. 9.000.
Fotocomposizione: Composit.
Stampa: Garzanti,
Milano. Distribuzione: SO.DI.P. Angelo
Patuzzi srl, Via Zuretti 25, Milano.
Registrato Trib. Milano N. 52 del 2/2/85.
Resp. Sira Rocchi.
Sped. in abb. post. Gr. III/70.

IN QUESTO NUMERO

MSX TAPE SOFT

RALLY
(RUN "CAS:RALLY")
di N. Paggin

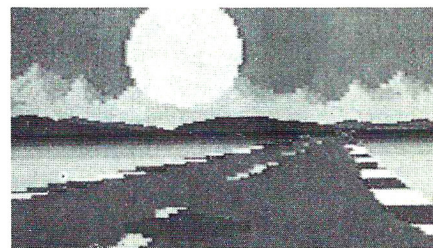
I programmi contenuti in questo fascicolo di MSX COMPUTER MAGAZINE sono tutti compatibili con qualsiasi sistema MSX. Ecco per voi ancora 10 bei programmi! Ricordate di collegare la spina del controllo motore alla presa REM del vostro registratore se quest'ultimo la possiede. Assicuratevi che la spina nera sia collegata alla presa EAR del registratore e che la spina rossa sia inserita nella presa MIC. Se il vostro mangiacassette non possiede la presa REM, fate particolare attenzione a quando un programma è stato caricato o deve esserlo, affinché il nastro scorra per il giusto tempo. Appena vedete apparire sul video, dopo un comando di caricamento, la scritta OK, spegnete il regi-

stratore. Nelle istruzioni che seguono troverete, accanto al titolo ed al codice di caricamento di ogni programma, la notazione della memoria minima necessaria per vedere il programma.

Nella cassetta allegata a questo fascicolo troverete, sul lato A: Rally, Bowling, Mixion, Corso d'inglese, Monitor.

Sul lato B ci sono: West, Mastermind, Puzzlemania, Sword & Spell, Copiatutto.

Immergiamoci nell'atmosfera dei grandi rallies del campionato del mondo con le sue prove speciali combattute sul filo del secondo e sui suoi fondi stradali più vari.

Il gioco è per un minimo di 2 giocatori fino ad un massimo di 8 giocatori.

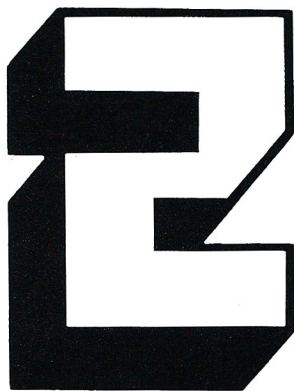
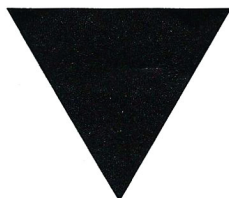
Le prove speciali si possono disputare su tre fondi stradali diversi: asfaltato, sterrato e ghiacciato. Ovvio che l'aderenza della vettura diminuirà in proporzione. In curva la macchina tende ad essere portata all'esterno dalla forza centrifuga e quindi sta a voi riuscire a tenerla in strada alla velocità più alta possibile per guadagnare secondi sugli avversari. Evitare il più possibile le uscite di strada: vi fanno perdere molto tempo e spezzano il ritmo!

Come nei veri rallies avete al vostro fianco un fido navigatore che vi preannuncia il tipo di curva che seguirà, o

la lunghezza del tratto rettilineo successivo, con una scritta sopra l'indicatore della velocità.
Esempio:
DESTRA 2 = curva media a destra
SINISTRA 3 = curva veloce a sinistra
DESTRA 1 = curva lenta a destra
RETTILINEO 250 = rettilineo di 250 m.

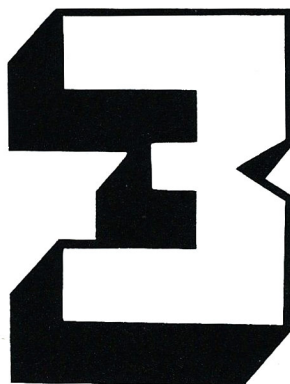
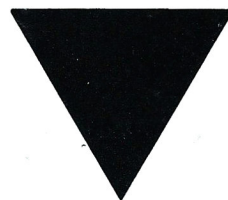
Un segnale acustico verso la fine di ogni rettilineo vi annuncerà l'avvicinarsi della curva successiva. Se i freni sono in buona efficienza, il tempo dovrebbe essere sufficiente a farvi rallentare quanto basta.

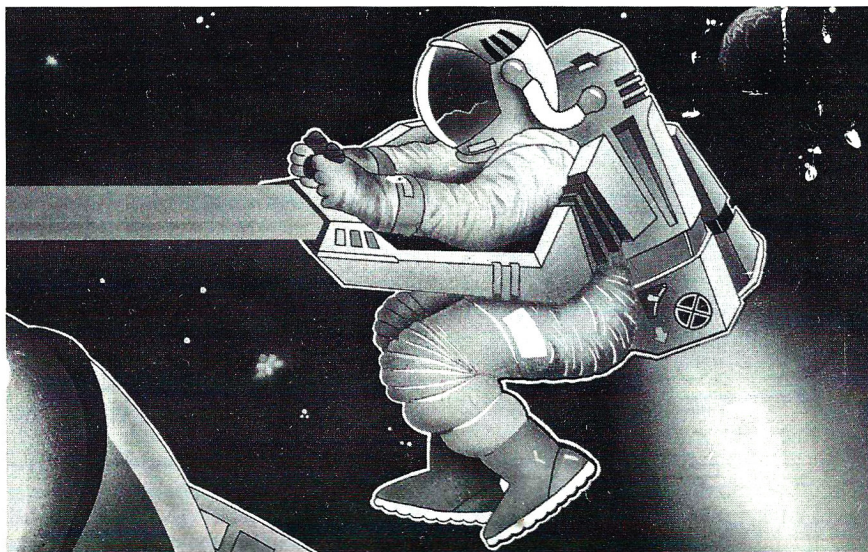
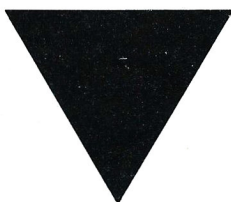
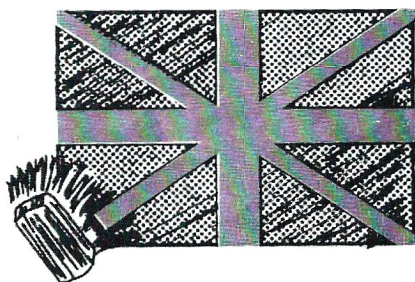
I freni ovviamente si consumano un po' ogni volta che vengono sollecitati. Il loro stato di efficienza all'inizio di ogni Prova Speciale è indicato in alto a destra sullo schermo. I freni vengono rimessi in efficienza solo ogni 4 Prove Speciali.

Si può usare sia il joystick che la tastiera. Per spostare la macchina a destra o a sinistra bisogna usare i tasti cursore corrispondenti. Il tasto cursore SU serve per frenare, quello GIÙ per accelerare.

ATTENZIONE: il programma va caricato con RUN "CAS:RALLY". A termine caricamento dare nuovamente RUN.

BOWLING
(CLOAD "BOWLIN" - 16 K)
di A. De Peppo

Avete mai provato a giocare a bowling? È davvero divertente così, se non avete mai provato, ecco l'occasione buona per iniziare.

Dopo aver dato il RUN dovrete scegliere se usare il joystick o la tastiera digitando la lettera corrispondente all'iniziale della periferica (T per tastiera e J per joystick).

Comparirà quindi una pista color viola, con dei birilli posti in fondo ad essa. Si gioca in due. A turno si deve scegliere la potenza (in una scala da 0 a 9) da imprimere alla palla che dovrà abbattere i birilli. Successivamente bisognerà selezionare la direzione della palla (destra D o sinistra S). A questo punto comparirà un omino all'inizio della pista. Spostatelo con i tasti cursore su e giù poi, scelta la posizione di tiro, premete il tasto di spazio. Vincerà chi totalizzerà il maggior punteggio, il quale dipenderà dal numero di birilli abbattuti. ■

MIXION
(CLOAD "MIXION" - 32 K)
di D. Montresor

Corre l'anno 2243; un astronauta in missione di esplorazione nel centro della galassia si avvicina ad un pianeta sconosciuto per studiarne la formazione. Ma ad un certo punto la nave da esplorazione si disintegra a contatto con l'atmosfera causa un guasto ai razzi frenanti. Il pilota si salva atterrando su un'isola in mezzo ad un fiume. Guidato dai vostri comandi, il pilota dovrà cercare i tre pezzi che compongono un traspositore di materia, in modo da poter tornare sulla Terra.

I comandi vengono impartiti nella seguente forma: verbo, articolo (facoltativo) ed oggetto oppure, in alcuni casi, solo dal verbo. Per muoversi nelle quattro direzioni basta scrivere l'iniziale del punto cardinale. Ecco l'elenco dei verbi o comandi speciali:
VERBI = permette di sapere quali azioni si possono compiere;
INVENTARIO = fornisce l'elenco degli oggetti che si hanno con sé;
AIUTO = fornisce dei consigli utili (funziona solo qualche volta!);
SALVO = salva su nastro la situazione alla quale siete arrivati nella avventura;
RECUPERO = carica da nastro un file registrato con il comando SALVO. ■

CORSO DI INGLESE
PARTE III
(CLOAD "PARTE3" - 32 K)
di G. Bellomusto

Continua il corso d'Inglese. Questa volta si impareranno le seguenti strutture grammaticali:
pronome oggetto;
CAN (verbo potere);
GIVE+Pronome oggetto + nome;
C'è — Ci sono;
imperativo;
SOME — A LOT OF;
ANY;
Verbo avere (affermativo);
Verbo avere (interrogativo);

la scelta di ogni argomento è fatta dal menù principale digitando il numero corrispondente. Quindi il programma

visualizza sullo schermo le regole grammaticali inerenti all'argomento scelto. A questo punto, alla domanda "Vuoi un commento (s/n)", possiamo scegliere di avere una descrizione più dettagliata sull'argomento. Se invece passiamo oltre, possiamo fare degli esercizi per valutare il nostro grado di apprendimento.

L'intera lezione si conclude con l'opzione E, esercizi vari.

MONITOR MSX
(CLOAD "MONMSX" - 32 K)
di D. Montresor

Permette di scrivere e leggere nelle memorie RAM e VRAM dei computer MSX con almeno 32 KRAM utente, con qualsiasi tipo di codice. Dando il RUN appaiono le seguenti opzioni:

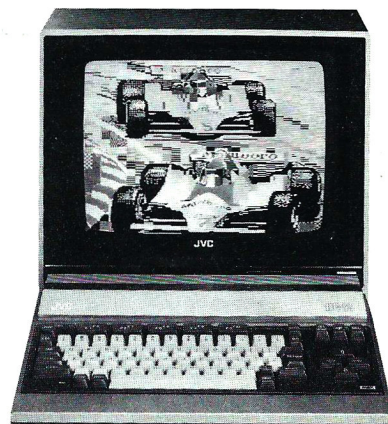
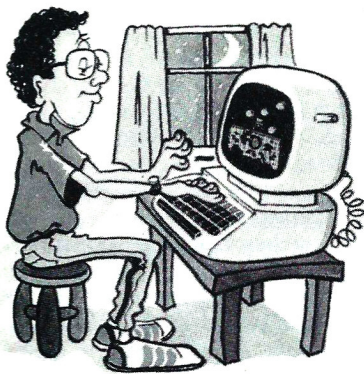
1) Visualizza la memoria; bisogna scegliere nel sottomenù se RAM o VRAM poi specificare l'indirizzo di partenza in decimale (per indirizzi esadecimali basta scrivere prima il prefisso "&H") e quindi vengono visualizzate 24 locazioni di memoria a partire dall'indirizzo specificato. A sinistra ci sono gli indirizzi delle locazioni in decimale e in esadecimale, quindi il contenuto delle stesse in binario, decimale, esadecimale, ottale ed in codice ASCII (i codici inferiori a 32 vengono convertiti nei caratteri grafici speciali). Gli unici caratteri che non possono venire normalmente visualizzati sono: 0, 32, 127; per vedere il seguito si preme un tasto e per tornare al menù SELECT.

2) Permette di scrivere nella memoria con qualsiasi codice senza prefisso, sia RAM che VRAM specificando, anche in questo caso, l'indirizzo di partenza che per la RAM può anche essere negativo (da —32768 a +65535) in quanto —32768 corrisponde a +32768 e —1 a +65535.

La memoria disponibile per il linguaggio macchina parte da 42500 ed arriva a 62336 (il CLEAR si trova all'inizio del programma e può essere modificato a piacere visto che non vengono utilizzate routine in L/M).

Durante l'input della memoria viene visualizzato, oltre che l'indirizzo in decimale ed esadecimale, il contenuto della locazione in cui si vuole immettere un valore in binario, decimale ed esadecimale.

Per uscire è necessario premere il tasto a (chiocciola) seguito da RETURN, mentre per confermare il contenuto della locazione visualizzata e passare alla successiva basta premere RETURN.

3) Conversioni: si sceglie il formato del valore da convertire e poi quello del risultato. Per tornare al menù premere ENTER senza scrivere niente.

4) Con questa opzione si può registrare una parte della RAM con il normale BSAVE oppure una parte della RAM o VRAM sotto forma di programma di linee DATA completo di ciclo FOR-NEXT caricatore.

Nel secondo caso, il programma così registrato può essere caricato con LOAD"CAS:" oppure fuso con un altro programma con le funzioni MERGE e RENUM.

5) Salto a subroutine in linguaggio macchina specificando l'indirizzo di partenza; il salto avviene dopo avere premuto un tasto.

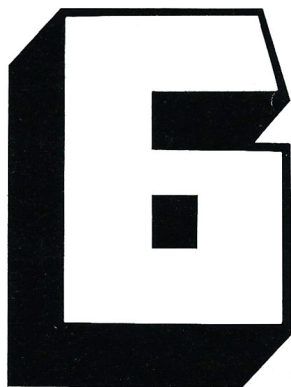
6) Recupero di una parte di RAM con BLOAD.

7) È possibile copiare un blocco di RAM o VRAM da una zona ad un'altra. Bisogna fornire l'indirizzo di partenza del blocco da copiare, l'indirizzo dove si intende trasferirlo e, per finire, la lunghezza dello stesso.

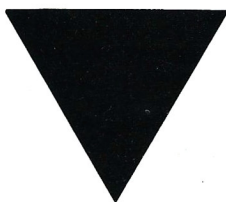
Non si può copiare un blocco ad un indirizzo che si trovi più vicino della lunghezza dello stesso, ma si può ovviare con due trasferimenti successivi.

8) L'opzione 8 del menù principale permette di ottenere il disassemblato di routine in L/M specificando l'indirizzo di partenza delle stesse; per fermare il listato usare il tasto STOP. Nel caso il disassemblatore incontri un'istruzione errata, appare apposito messaggio: questo può succedere se l'indirizzo dato all'inizio non coincide con il vero indirizzo della routine, oppure se la zona di memoria esaminata è riservata a delle variabili.

Per tornare al menù principale premere il tasto SELECT, valido per ogni opzione.

■

WEST

(CLOAD "WEST" - 32 K)

di Massimo Belardi

Il programma si divide in tre fasi: duello, agguato e caccia.

Per giocare si usa il joystick o i tasti cursore (il programma controlla entrambi contemporaneamente).

Spostando la leva a destra o a sinistra si sposta il pistolero in queste direzioni; spostandola in alto o in basso, si sposta in queste direzioni una piccola pistola che stabilisce il punto in cui cadrà il proiettile quando si spara. Per sparare si deve prima spostare la leva in alto o in basso per stabilire "l'alzo" (infatti solo così il pistolero estrae la pistola), poi si preme il pulsante di fuoco o la barra spaziatrice.

DUELLO - Si deve estrarre e sparare con precisione verso il pistolero che è uscito dal Saloon; esso sparerà nello stesso istante in cui voi estrarrete la pistola. Nel primo round il pistolero è uno, ma ne troverai via via altri fino ad un massimo di 4. Per ogni pistolero colpito si vincono 100 punti.

AGGUATO - Da dietro le rocce escono pistoleri che sparano all'impazzata cambiando continuamente posizione e frequenza di tiro. Nel primo round sono tre ed attaccano uno alla volta; nel secondo sei ed attaccano due alla volta; nel terzo sono nove ed attaccano tre alla volta. Ad ogni round aumentano di tre ed attaccano sempre in tre contemporaneamente. Per ogni pistolero colpito si vincono 10 punti.

CACCIA - In questo quadro al nostro pistolero non spara nessuno; egli deve cercare di colpire più bufali possibile prima che escano dallo schermo. Sono presenti contemporaneamente sei bufali e ogni volta che ne viene abbattuto uno, oppure quando un bufalo esce dallo schermo, ne arriva un altro. Nel primo round sono 10 e aumentano di 10 ad ogni round fino ad un massimo di 70. Se si colpiscono tutti prima che escano dallo schermo, si otterrà un bonus di 1000 punti. Per ogni bufalo si ricevono 10 punti alla fine della caccia (o 1000 se sono stati colpiti tutti).

Si comincia con tre vite e se ne vince una alla fine di ogni round per un massimo di sei. Alla fine del gioco verrà chiesto il nome del giocatore, che verrà messo in graduatoria nel tabellone dei punteggi.

Per iniziare a giocare bisogna premere la barra spaziatrice o il tasto fire del joystick.

Il programma si compone di diverse routine in linguaggio macchina. La routine che controlla il joystick si trova all'indirizzo B100 fino all'indirizzo B22F.

Per caricare il programma digitare CLOAD "WEST" e poi, una volta apparsa la scritta "OK", digitare RUN per il caricamento del linguaggio macchina.

■

7

MASTERMIND
(CLOAD "MASTER" - 16 K)
di Paolo Sellan

Ogni tanto fa piacere tornare a giocare i classici videogame come questo mastermind, gioco assai famoso di deduzione logica.

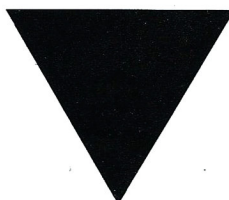
Caricato il programma, date RUN e scegliete, premendo il tasto corrispondente, se usare la tastiera o il joystick. Scopo del gioco è riuscire a trovare la corretta combinazione formata da 5 palline colorate, combinazione scelta dal computer. Per giocare bisogna spostare, utilizzando i tasti cursore o il joystick, un quadratino su una serie di palline colorate e formare la combinazione da trovare premendo il tasto fire del joystick o quello di spazio della tastiera. In caso di scelta errata è possibile cancellare l'errore portando il quadratino sulla lettera C e digitando il tasto di spazio. Se invece le cinque palline scelte possono rappresentare la combinazione giusta, allora portate il quadratino sopra la lettera X. A questo punto il computer valuta la risposta e restituisce dei pallini vuoti, tanti quante sono le palline di colore esatto ma poste, nella sequenza, in posizione errata e dei pallini pieni, tanti quante sono le palline di colore e posizione esatta.

Per indovinare la combinazione si hanno a disposizione fino a 10 tentativi!

■

PUZZLE MANIA
(CLOAD "PUZMAN" - 32 K)
di E. Marcon

Ecco un ottimo programma di utilizzo degli sprite. Grazie al loro facile utilizzo è possibile spostarli in ogni parte del video creando una tale confusione da ricordare le immagini spezzate dei puzzle.

Dopo la sigla iniziale compare una videata con la possibilità di scelta tra tastiera o joystick e gioco o istruzioni.

In seguito alle istruzioni ecco il menù dal quale scegliere tra 5 schemi diversi oppure se terminare il GIOCO.

I 5 schemi sono ordinati per difficoltà di composizione (al numero più alto corrisponde lo schema più difficile). Ecco quindi innanzitutto la prima immagine completa della figura da comporre (soluzione), quindi i tasselli mescolati secondo 5 possibilità (vedere righe 7530 in poi) proseguendo nel gioco vi accompagnerà una musica di sottofondo.

I tasti comando sono:
<space> per controllare la propria posizione
<esc> per tornare al MENÙ in qualsiasi momento del gioco.

In qualche caso, sebbene la posizione scelta sia esatta, non verrà dato l'"OK" centrale, perché il tassello non è quello giusto. In questo caso compare un "NO". A questo punto si può: richiamare il tassello per cercare un'ulteriore posizione, oppure lasciare tale tassello in quella posizione per cercare di piazzarne un altro.

In caso di completamento esatto finale, compare una sigla grafica e musicale, che rimanda al MENÙ.

Possibili modifiche al programma: potete cambiare i DATA dei vari schemi per nuove figure, oppure inserire la riga '7610 s=int(rnd(—time)*4)+1' per la scelta casuale della posizione iniziale di ogni tassello.

SWORD & SPELL II
(CLOAD "S&S II" - 32 K)
di Stefano Guarnaschelli

Dopo essere riuscito a fuggire dal labirinto del mago (Vedi S&S n. 1 sul fascicolo n. 11) è ora giunto il momento di passare al contrattacco colpendo il malvagio nel suo castello. Spetta a te utilizzare al meglio i tuoi

poteri magici, la tua spada e gli oggetti che troverai nelle sale del palazzo, per portare a termine la tua missione e sconfiggere una volta per tutte il tuo nemico salvando così la principessa Elisa da un'eterna prigionia.

Ecco come si gioca:

1) MOVIMENTO: la direzione di movimento viene scelta tra quelle possibili (non è possibile attraversare i muri!) tramite i tasti cursore o il joystick; non è possibile attraversare una stanza nella quale sia presente un avversario senza averlo prima eliminato.

2) COMBATTIMENTO: premendo il tasto F1 si potrà scegliere uno dei modi di attacco (spada, fulmine, sfera di fuoco) posizionando la crocetta sul simbolo corrispondente e premendo la barra spaziatrice o il pulsante di fuoco; utilizzando gli incantesimi vengono "bruciati" dei punti di vitalità; attaccando con la spada, i danni inferti e subiti saranno proporzionali allo status dei contendenti.

Qualora si sia cercato di attaccare in modo errato, con gli incantesimi o con la spada, un avversario ad essi immune, una volta subita la sua reazione, si potrà tornare alla opzione di scelta degli oggetti posizionando la crocetta in basso a sinistra e premendo la barra o il pulsante di fuoco; in tal modo si potrà, tramite i tasti funzione, passare alla scelta dell'oggetto adatto.

3) USO DI OGGETTI: alcuni mostri risultano essere immuni ai normali attacchi dell'eroe; contro di essi sarà necessario utilizzare gli oggetti precedentemente recuperati; ciò sarà

possibile premendo il tasto F2 e selezionando l'oggetto tramite i tasti cursore, la crocetta, la barra spaziatrice o il tasto di fuoco.

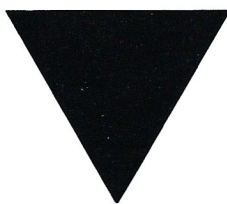
Per ritornare alla possibilità di scelta di movimento o di combattimento, si dovrà posizionare la crocetta in basso a destra e premere la barra o il pulsante di fuoco. Attenzione: qualora si scelga un attacco o un oggetto errato il mostro ne approfitterà per scagliarsi con violenza contro l'eroe infliggendogli danni superiori al normale.

4) POSIZIONE RISANANTE: all'inizio dell'avventura il nostro eroe dispone di 15 dosi di posizione utilizzabili selezionando il modo di uso degli oggetti e scegliendo il simbolo (croce) in alto a sinistra: in tal modo verranno curate alcune ferite.

Alcuni mostri sono colpibili unicamente utilizzando specifici oggetti:

Tavashri: con la pergamena.
Stregone: con l'anello.
Spettro: con la croce.
Fred: con l'anfora.
Budino: con il cucchiaio.
Licantropo: con il pugnale.

Altri mostri sono refrattari ad alcuni tipi di attacchi:

Drago: alla sfera di fuoco.
Demone: al fulmine.
Cavaliere Rosso: alla spada.
Cavaliere Giallo: al fulmine.
Cavaliere Blu: alla sfera di fuoco.

10 COPIA TUTTO

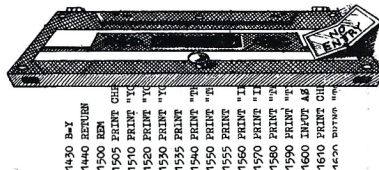
(CLOAD "COPIAN" - 16 K)

di Marco Dal Monte

Ecco un programmino breve breve ma assai potente. Uno di quei programmi il cui uso deve essere consacrato unicamente alla messa a posto della propria softeca e... Se non avete ancora capito di cosa si tratta, stiamo parlando di un programma capace di eseguire copie di qualunque software, sia esso in Basic, in linguaggio macchina o anche una semplice videata grafica, insomma qualsiasi file registrato su nastro, anche quelli senza header!.

Nel caso in cui il programma da copiare sia (come quasi sempre succede) suddiviso in più parti, il copiatore ne caricherà una per volta per poi riversarla sulla cassetta "pirata". Essendo ogni parte di un programma strutturata in modo differente, il copiatore necessita di alcune informazioni basilari per eseguirne il caricamento ed il riversamento in modo corretto. Per questo il programma, una volta partito, mette a disposizione 6 possibilità di scelta sul tipo di file da copiare ed è qui che forse si potrebbero presentare delle difficoltà per chi non fosse esperto. Con il termine HEADER, che appare in tutte le opzioni, si intende quel brevissimo programmino che si può sentire prima del programma vero e proprio, che contiene i dati inerenti al programma che il computer sta per caricare. Questo Header può essere facilmente individuato con l'ascolto audio del programma che si intende copiare. Nel caso l'Header sia assente, optate per una qualunque altra opzione che non necessiti dell'Header per poterne effettuare la copia.

Essendo l'Header considerato dal copiatore come non facente parte integrante del programma, viene carica-

to e subito riversato sul secondo nastro senza che venga caricato il programma vero e proprio, a meno che questo non sia in Basic: in questo caso il programma verrà copiato integralmente. Attenzione: questo copiatore non può convivere in memoria con altri programmi BASIC, salvo accorgimenti particolari.

Per caricare il programma digitare CLOAD "COPIAN" e poi dare il RUN per il caricamento del linguaggio macchina.

■

GRAPHIC

EIDOMATICA ABC

IL FASCINO DELL'ELABORAZIONE GRAFICA SUL COMPUTER È INNEGABILE. LE PRIME ROUTINES PER IMPADRONIRSI DI UNA TECNICA CHE SARÀ ESSENZIALE PER MOLTE PROFESSIONI DI DOMANI

a cura della Redazione

L'eidomatica, quella parte d'informatica che si occupa della immagine, è senz'altro una delle applicazioni più interessanti nel mondo dei computer. Con questo articolo vogliamo iniziare a far conoscere, con applicazioni pratiche, il fascino dell'elaborazione grafica scoprendo di volta in volta le varie tecniche di programmazione. Incominceremo con le rappresentazioni grafiche in bidimensione per arrivare poi alle complesse immagini tridimensionali. Ma prima ancora di parlare di grafica avanzata dobbiamo aver chiare le prestazioni grafiche di un MSX.

Lo schermo grafico dell'MSX è formato da una matrice rettangolare di punti indirizzabili individualmente detti pixel. Più precisamente abbiamo 256 colonne per 192 righe di punti. Ogni unità elementare grafica (il pixel) può essere identificata da una coppia d'interi (X,Y) detta anche vettore pixel dove naturalmente le variabili x e y vanno definite in un certo intervallo.

0<=X<=255 e 0<=Y<=191

Il vettore (0,0) indica il pixel più in alto a sinistra.

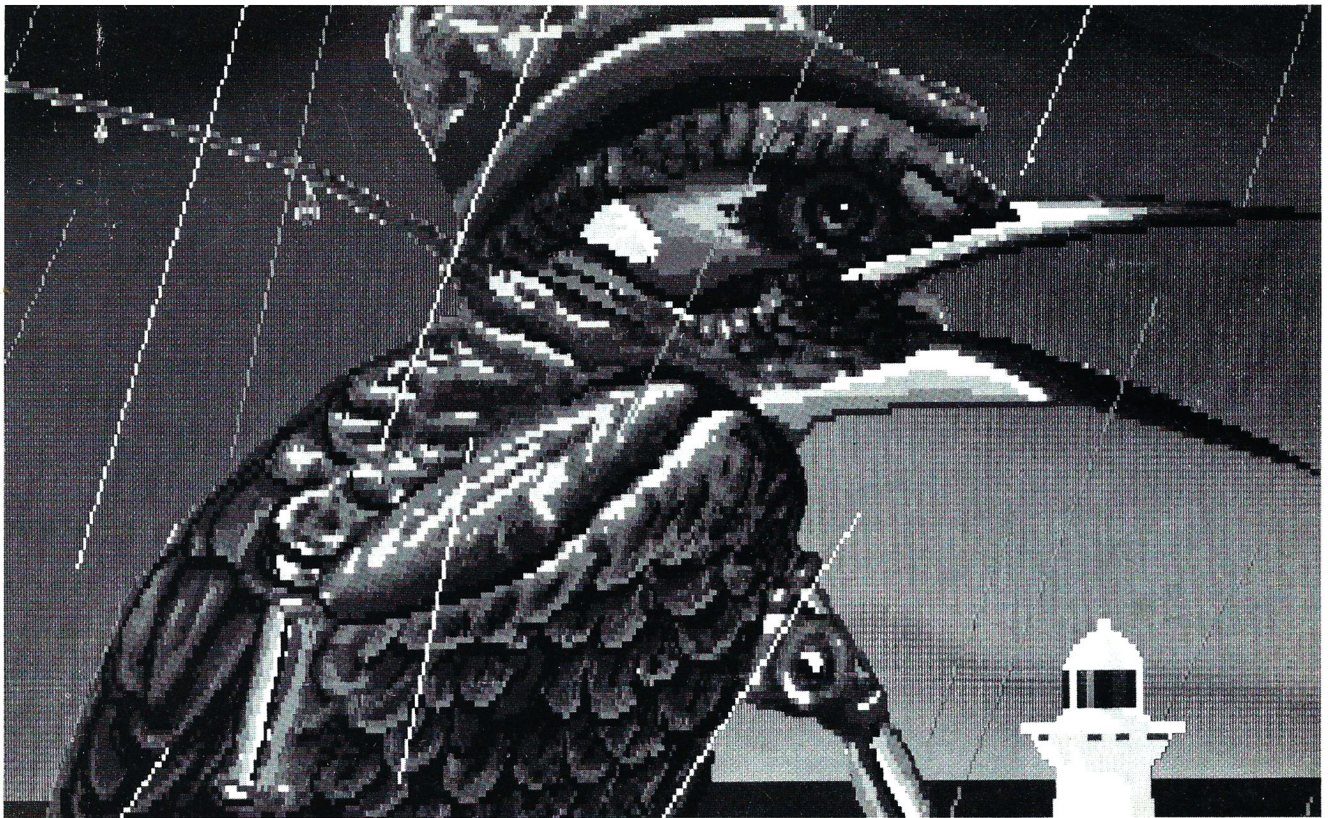
Essendo il vettore pixel individuato da due parametri la trattazione delle figure bidimensionali sarà quasi immediata.

GRAFICA BIDIMENSIONALE

Lo spazio bidimensionale ha come primo riferimento le coordinate carte-

UN PROGRAMMA PER UN QUADRATO

```
10 DEF FNX(C)=INT(XO+(C*XYSC)+.5)
20 DEF FNY(C)=INT(YO+(C*YXSC)+.5)
30 SCREEN 2
40 XYSC=1:YXSC=1
50 XO=127:YO=96
100 REM disegno di un quadrato
159 REM unisce i vertici del quadrato
160 XPT=50:YPT=50:GOSUB 9500
170 XPT=40:YPT=50:GOSUB 9400
180 XPT=40:YPT=40:GOSUB 9400
190 XPT=50:YPT=40:GOSUB 9400
200 XPT=50:YPT=50:GOSUB 9400
210 GOTO 210
9400 REM traccia linea
9410 NX=FNX(XPT)
9420 NY=FNY(YPT)
9430 LINE (XC,YC)-(NX,NY)
9440 XC=NX:YC=NY
9450 RETURN
9500 REM sposta a XPT,YPT
9510 XC=FNX(XPT)
9520 YC=FNY(YPT)
9530 RETURN
```

Un semplice esempio di utilizzo delle subroutine necessarie per... disegnare. Sullo schermo, un quadrato!

siane ovvero due rette che procedono all'infinito e tra loro perpendicolari. La retta che procede orizzontalmente viene detta asse delle x mentre la retta che procede verticalmente asse delle y. Il punto d'incrocio di queste due rette si chiama origine delle coordinate. Su tali assi definiamo una unità di misura che può essere anche differente tra i due assi ma che però assume valori positivi, per l'asse x, alla destra dell'origine e valori negativi alla sinistra dell'origine. Per l'asse y, invece, i valori positivi sono quelli al di sopra dell'origine mentre quelli negativi al di sotto. Da queste definizioni possiamo distinguere univocamente un qualsiasi punto P dello spazio bidimensionale attraverso le sue coordinate o vettore bidimensionale (XP, YP). Ora possiamo trasportare il punto geometrico P (XP, YP) sullo schermo prendendo come spazio bidimensionale lo schermo stesso. Più precisamente lo schermo rappresenterà una porzione dello spazio in quanto non è possibile far rientrare tutti i punti dello spazio in un'area definita. Secondariamente dobbiamo definire l'origine sullo schermo dello spazio che sarà dipendente dai grafici che vogliamo rappresentare, ovvero potrà essere centrata nel video oppure tutta da un lato.

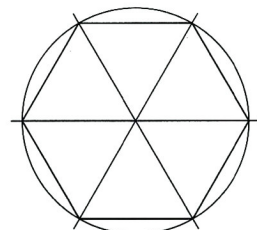
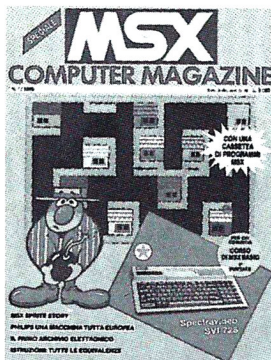
In precedenza abbiamo detto che la larghezza in x dello schermo grafico dell'MSX è di 256 pixel mentre quella in y è di 192 pixel; molto spesso capita che tali valori sono insufficienti a rappresentare l'intero grafico. Si rende così necessaria una routine per convertire i dati reali delle coordinate dei punti in coordinate dei pixel, perciò introduciamo le due funzioni FNX e FNY:

```
DEF FNX(C) =
INT(XO+(C*XYSC)+.5)
```

```
DEF FNY(C) =
INT(YO+(C*YXSC)+.5)
```

dove il parametro C passato alla funzione indica il valore della coordinata reale mentre XO e YO sono le coordinate dell'origine sul video e XYSC è il fattore di scala sull'asse x, cioè il numero massimo di pixel in orizzontale/massima lunghezza reale orizzontale e YXSC è il fattore di scala verticale ovvero numero massimo di pixel verticali/max lunghezza reale verticale.

SE SI VUOLE UN POLIGONO 2

```
300 REM POLIGONO
310 XPT=X(NV):YPT=Y(NV):GOSUB 9500
320 FOR I=1 TO NV
330 XPT=X(I):YPT=Y(I):GOSUB 9400
340 NEXT I
350 RETURN
```

Una routine dedicata per il disegno di poligoni con n vertici. Per evidenziare i lati si sfrutta la subroutine traccia linee.

Supponiamo per esempio di aver definito l'origine del nostro sistema di riferimento nel punto (127,—96) (abbiamo scritto —96 per spostare la coordinata y dell'origine in basso in modo da individuare l'origine dello schermo dell'MSX anziché in alto a sinistra in basso) e di voler trovare le coordinate del pixel rappresentante il punto reale (10,10) posizionato sull'estrema destra in alto al video, calcoleremo:

```
XReale =
INT(127+(10*256/20)+.5) = 255
YReale =
INT(—96+(10*192/20)+.5) = 0
```

infatti il vettore (255,0) rappresenta il pixel più in alto a destra.

I PRIMI STRUMENTI GRAFICI

Per incominciare a disegnare le prime figure bidimensionali abbiamo bisogno di almeno altre due routine: una che sposti il cursore grafico senza tracciare e l'altra che lo sposti tracciando una linea.

La prima routine, che chiameremo "sposta", verrà scritta come segue:

```
9500 REM sposta a XPT, YPT
9510 XC = FNX (XPT)
9520 YC = FNY (YPT)
9530 RETURN
```

I dati di input sono XPT e YPT cioè le coordinate grafiche dove spostarsi. I valori di input vengono poi adattati ai valori del pixel corrispondente, usando le funzioni FNX e FNY descritte in precedenza, e registrate in XC e YC che sono le coordinate del cursore grafico.

La seconda routine in questione, che chiameremo "traccia linea", disegna una linea, dalla posizione corrente del cursore grafico al punto P individuato dal vettore (XPT, YPT):

```
9400 REM traccia linea
9410 NX = FNX (XPT)
9420 NY = FNY (YPT)
9430 LINE (XC, YC) (NX, NY)
9440 XC = NX: YC = NY
9450 RETURN
```

Anche questa routine sfrutta le funzioni FNX e FNY per ricavare i valori del punto P adattati allo schermo grafico. Si noti alla linea 9430 l'utilizzo della prima istruzione grafica LINE che collega appunto due pixel di date coordinate. Successivamente alla linea 9440 vengono aggiornate le coordinate (XC, YC) del cursore grafico all'ultimo punto tracciato.

UN ESEMPIO

Il listato 1 è un semplice esempio di utilizzo delle subroutine fin qui spiegate e disegna un quadrato di lato 10, ovviamente adattando la scala di risoluzione.

Il listato 2, invece, è la routine dedicata apposta al disegno di poligoni aventi n vertici. Quest'ultima routine ci servirà più avanti. Intanto spieghiamo qui di seguito come funziona.

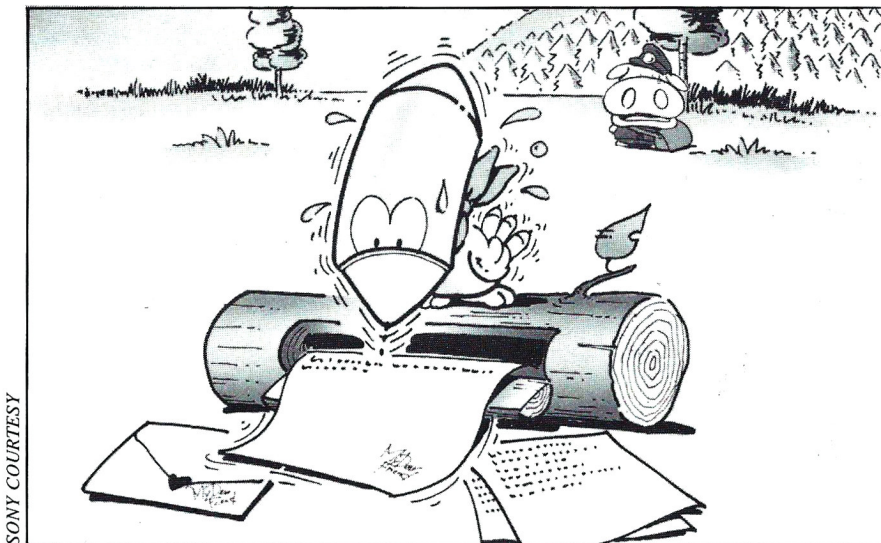
I dati di input sono i vettori X(),Y() e la variabile NV indicante il numero dei vertici del poligono. Alla linea 310 vengono preparate le variabili da passare alla subroutine "sposta" per il posizionamento del cursore grafico e poi tramite un ciclo FOR-NEXT da 1 a NV, vengono prelevate tutte le coordinate dei vertici del poligono che serviranno a tracciare, sfruttando la subroutine "traccia linea", i lati della figura. ■

SOFTWARE

WORD PROCESSOR

ORGANIZZAZIONE GENERALE DI UN PROGRAMMA DI ELABORAZIONE TESTI. STESURA, STRUTTURA E LISTATO COMMENTATO

di EMANUELE DASSI

SONY COURTESY

Saper programmare non è facile, è un'arte rigorosa che non lascia spazio all'irrazionale però è pur vero che si può risolvere lo stesso problema in diversi modi. Ed è per questo che il risultato di una elaborazione può essere raggiunto in tempi differenti, con una tecnica più o meno elegante, adoperando quantità di memoria diverse. Stendere l'algoritmo migliore richiede una corretta analisi del problema ed una buona conoscenza dell'elaboratore che dovrà ospitare il programma.

Ogni programmatore vorrebbe certamente migliorare il proprio modo di usare il computer ma spesso non ha la possibilità di frequentare una scuola o semplicemente non sa a chi rivolgersi con la conseguenza di limitarsi ad usare il proprio computer con programmi acquistati o programmando da sé semplici algoritmi. Con gli articoli che via via pubblichiamo vogliamo cercare di trasmettere al lettore le nostre esperienze di programmatori affinché possa aumentare il suo grado di preparazione.

L'arte della programmazione richiede tanta sperimentazione quindi non vi è cosa migliore che apprenderla tramite la stesura di un programma. Consideriamo che il lettore conosca già il Basic dell'MSX e qualche nozione di base di Assembler Z80. Il progetto che vogliamo realizzare è la programmazione di un WP.

CHE COS'È UN WP

WP è l'abbreviazione dei termini Word-Processor, ovvero Elaboratore di Testi. Un programma del genere serve appunto ad elaborare dei testi, cioè permette di utilizzare il proprio computer come una macchina da scrivere. Usando un WP è possibile scrivere lettere, pagine di un libro, tabelle e così via; quanto scritto viene visualizzato sul video, memorizzato su cassetta o disco e stampato su carta tramite una stampante. Un WP si compone di più routine che aiutano l'utente nella stesura del testo e delle quali parleremo più avanti.

L'elaboratore testi è quindi un programma molto utile e certamente non può mancare nella softeca di un programmatore. Implementare un WP su di un MSX vuol dire realizzare il massimo delle prestazioni sopra elen-

LA GESTIONE DATI

```
1000 RK$=INKEY$:IF RK$=""THEN 1000
1010 CRK%=ASC(RK$)'ESTRAE VALORE ASCII
     DEL CARATTERE RK$
1020 IF CRK%=13 THEN GOSUB 1200:RETURN '
     CARATTERE ENTER
1030 IF CRK%>31 AND CRK%<255 AND CRK%<>1
27 THEN GOSUB 1200:RETURN ' CARATTERE
     ALFANUMERICO
1040 IF CRK%=8 AND LM>50000! THEN LM=LM-
1:POKE LM,32:IF LM>=49999!+LT% THEN LT%=
LT%-1:RETURN ELSE RETURN 'TASTO      BS
1050 IF CRK%=127 AND LT%>0 AND LM<=49999
!+LT% THEN LT%=LT%-1:FOR NN=LM TO 49999!
+LT%-1:POKE NN,PEEK(NN+1):NEXT NN:POKE N
N,32:RETURN ' TASTO DELETE
1060 IF CRK%=27 THEN GOSUB 1400:RETURN '
     TASTO ESCAPE - HELP
1070 IF CRK%=24 THEN GOSUB 4000:RETURN '
     TASTO SELECT - COMANDI VARI
1080 IF CRK%=30 AND LM>=50000!+LP% THEN
LM=LM-LP%:RETURN 'TASTO FRECCIA SU
1090 IF CRK%=28 AND LM<49999!+LT% THEN L
M=LM+1.RETURN 'TASTO FRECCIA DESTRA
1100 IF CRK%=31 AND LM+LP%<=49999!+LT% T
HEN LM=LM+LP%:RETURN 'TASTO FRECCIA
    GIU'
1110 IF CRK%=29 AND LM>50000! THEN LM=LM
-1 'TASTO FRECCIA SINISTRA
1120 RETURN
1200 POKE LM,CRK%:LM=LM+1:IF LM>49999!+L
T% THEN LT%=LT%+1
1210 IF LM<62336! THEN RETURN
1220 CLS:LOCATE 10,10:PRINT "FINE MEMORI
A!":LOCATE 10,11:PRINT "SALVARE IL TESTO
!":LOCATE 5,12:PRINT"PREMI UN TASTO PER
CONTINUARE"
1230 RK$=INPUT$(1):RK$="":RETURN
```

La gestione dei dati immessi dall'utente.
Si tratta di una subroutine che si occupa dell'input da tastiera.
Si noti in particolare la linea 1200
ove si prevede di memorizzare come testo l'input stesso.

cate tenendo conto delle limitazioni del computer tra le quali le più importanti sono: risoluzione del video, memoria disponibile e linguaggio di programmazione.

STESURA DI UN WP

Incominciamo dalla risoluzione del video con la quale vogliamo intendere il numero massimo di caratteri visualizzabili su di una linea. Per l'MSX tale numero è 40, un po' poco per poter scrivere un testo normale di circa 60-80 caratteri per linea. Ecco quindi un primo problema: come fare a visualizzare più di 40 colonne? La soluzione potrebbe essere quella di far scorrere lateralmente il testo in modo da avere l'impressione che il video si estenda oltre le 40 colonne.

Un'altra limitazione è la memoria disponibile che nel nostro caso consideriamo di 32K RAM (pensiamo di sviluppare il programma su un computer MSX avente almeno tale quantità di memoria) nella quale dobbiamo far entrare il programma e l'area riservata alla memorizzazione del testo. Ricordiamo che scrivere un testo richiede molta memoria: una pagina da 60 colonne per 30 righe equivale ad occupare circa 1,7K di memoria. Per lasciare più memoria possibile alla memorizzazione del testo bisognerà scrivere un programma conciso che utilizzi il minimo indispensabile di variabili.

L'ultima importante limitazione è il linguaggio di programmazione che sarà il Basic MSX. Essendo quest'ultimo un linguaggio interpretato il programma richiederà tempi di elaborazione superiori a quelli di un linguaggio compilato (es.: Pascal) ma in compenso rispetto ad un programma in linguaggio macchina sarà più facile da scrivere.

STRUTTURA DI UN WP

Ed ora dopo aver considerato le limitazioni del mezzo sul quale sviluppare il programma vediamo di descrivere la struttura di un WP. Possiamo dividere il programma in tre punti principali:
1 — Input da tastiera
2 — Output su video
3 — Operazioni sul testo.

Per quanto riguarda il punto 3 questo comprende una serie di opzioni delle quali alcune sono sempre operanti mentre altre divengono attive solo dopo essere state richieste dall'utente. Elenchiamo tutte le opzioni:
3.1 - impostazione formato pagina: n. righe x n. colonne;
3.2 - paragrafo, centratura e marginatura;
3.3 - posizionamento diretto nel testo (GO);
3.4 - operazioni di blocco: copia ed eliminazione;
3.5 - operazioni su singola parola: ricerca e sostituzione;
3.6 - operazioni di stampa: numerazione pagine, intervallo tra linee, margine sopra e sotto;
3.7 - esecuzione della stampa;
3.8 - operazioni di archiviazione: save e load;
3.9 - abbandono programma.

Incominceremo con l'impostare la programmazione del punto 1: input da tastiera.

Questa parte di programma deve provvedere a gestire i dati immessi dall'utente, i quali possono essere:
— caratteri alfanumerici;
— movimento cursore nel testo nelle quattro direzioni;
— cancellazione di un carattere;
— tasto esecuzione comandi;
— tasto di help: visualizzazione comandi.

Inoltre l'algoritmo deve preoccuparsi di memorizzare i dati acquisiti.

ED ORA IL LISTATO

Il listato n. 1 contiene tutti i controlli sopra elencati. Vediamo di commentare le linee di programma.

```
"110800DD19216ED835C2DBD23A5ED8A7"
"281A219DD7CD27D70100000B78B120FB"
"325ED8CD24D7218BD7CD27D72A56D82B"
"7C[illegible]0FB[illegible]6D1[illegible]AD7[illegible]41C5"
"21    B7E    503    16B          60F"
"77    2D5    0F     3B1          CD"
"24    67     2E     DC1          0C"
"0B    D7     CI     711    216   3E"
"F5C   D      D      720    CDD   A7"
"28F   4      7      5DD    E60   31"
"13C   3   5  3      2802         DD"
"350      7          D8F5         28"
"011E         E      0528         112"
"D5C5         5      64D8    8AFCDD8"
"0032         7      A8112   D52164D8"
"3520         9A     A63D8   62D8CD22"
"D532         9D     2DD4E   22D5DD77"
"03C90        E60    2060F   87818180"
"C92A60D8110606 19E555CDFDD54BC521"
"89D816000061519 10FDC1D1E57BFE8038"
"02D6045FCDFDD5E1431600197EA7C8E5"
```

Innanzitutto va precisato che la parte di programma in esame è una subroutine che si occupa dell'input da tastiera. Per poter leggere qualsiasi tasto si è usata la funzione INKEY$ (linea 1000) la quale assegna alla variabile alfanumerica RK$ il carattere corrispondente al tasto premuto. A questo punto, se è stato premuto un tasto, il programma deve esaminare che tipo di carattere è, in base alle distinzioni descritte sopra. Per agevolare tutte le operazioni di confronto si assegna alla variabile intera CRK% il valore ASCII del carattere acquisito input (linea 1010). Prima di proseguire nella descrizione del listato dobbiamo dire che la subroutine utilizza tre variabili inizializzate prima che la subroutine venga chiamata, esse sono: LM, LT% e LP%. LM indica la locazione di memoria dove inserire il codice del carattere prelevato in input. Sulla base di questo valore vedremo più avanti che si ricostruiscono le coordinate X e Y del cursore nel testo.

L'area di memoria riservata alla memorizzazione del testo va dall'indirizzo 50000 all'indirizzo 62335. Il testo non viene memorizzato in array alfanumerici ma direttamente in memoria tramite l'istruzione POKE. L'utente ha quindi a disposizione circa 12KRAM equivalenti a poco più di 7 pagine da 60 colonne per 30 righe.

La variabile LT% contiene il numero di caratteri inseriti nel testo. È importante sottolineare la differenza tra LT% e LM, la prima indica la quantità di memoria occupata la seconda invece la posizione del cursore nel testo.

Infine vi è la variabile LP% indicante la larghezza della pagina, ovvero il numero di colonne.

Continuiamo la descrizione del listato. Le linee 1020-1030 controllano che il valore di CRK% corrisponda a quello di un carattere stampabile, compreso il tasto ENTER (valore ASCII=13) il cui effetto è quello di mandare a capo il cursore. Entrambe le linee eseguono l'istruzione IF/THEN il cui valore vero comporta l'esecuzione della subroutine alla linea 1200. Quest'ultima è la parte più importante di tutto il programma perché memorizza come testo l'input da tastiera. Le istruzioni POKE LM, CRK% e LM=LM+1 (linea 1200) servono a memorizzare nella locazione LM il valore ASCII del carattere digitato, contenuto nella variabile CRK%, e ad incrementare il puntatore nel testo. Successivamente (linea 1210) se il valore di LM è minore di 62336 ciò vuol dire che vi è ancora spazio nella memoria del testo altrimenti (linee 1220-1230) vengono visualizzati alcuni messaggi.

La cancellazione di un carattere è affidata alla linea 1040 la quale, dopo aver controllato che CRK% sia uguale a 8 (tasto BS) e che LM>50000, ovvero che sia stato inserito almeno un carattere, procede a decrementare il puntatore LM% e a cancellare l'ultimo carattere digitato inserendo al suo posto nella memoria il valore ASCII 32 corrispondente allo spazio; inoltre viene decrementato di uno il contatore della lunghezza del testo LT%.

Se CRK% non è uguale a 8 la linea 1050 controlla che non sia 127 (tasto delete), in tal caso se la lunghezza del testo (variabile LT%) è maggiore di 0 ed il cursore non è posizionato oltre la fine del testo inserito (LM%<=49999!+LT%) allora viene decrementata la lunghezza del testo e, utilizzando un ciclo FOR-NEXT, il testo viene accorciato di un carattere a partire dal primo sulla destra del cursore, realizzando la funzione delete.

La linea 1060 provvede a riconoscere la pressione del tasto ESCAPE (valore ASCII 27) e ad eseguire la subroutine alla linea 1400 (che scriveremo alla fine dell'intero programma) la quale visualizza in sintesi i comandi del programma. La linea successiva viene eseguita se è stato premuto il tasto SELECT ed esegue la subroutine 4000 (di cui parleremo prossimamente) contenente tutti i comandi attivabili dall'utente, ovvero le "operazioni sul testo". Infine dalla linea 1080 alla linea 1120 vengono eseguiti i controlli sulla pressione dei tasti delle frecce ed aggiornata la posizione del cursore variando il valore della variabile LM.

Riassumendo abbiamo visto la subroutine che si dedica esclusivamente all'input da tastiera, alla memorizzazione dei caratteri facenti parte del testo scritto dall'utente ed alla variazione di alcune variabili. A proposito di quest'ultime riportiamo qui di seguito l'elenco delle variabili ed il loro significato:

LT% = lunghezza del testo espressa in numero di caratteri

LM = locazione di memoria dove inserire il prossimo carattere del testo

LP% = larghezza pagina testo, cioè il numero delle colonne del foglio

RK$ = input da tastiera

CRK% = valore ASCII DI RK$

NN = contatore del ciclo FOR-NEXT di linea 1050

Diamo appuntamento alla prossima puntata dove continueremo la stesura del WP, intanto abbiamo visto come si imposta la risoluzione di un problema, dividendo l'intero lavoro in più parti e sviluppando mano a mano ognuna di esse.

Ⓛ Ⓞ Ⓖ Ⓘ Ⓒ

ALGEBRA BOOLEANA

GLI OPERATORI LOGICI SONO MOLTO IMPORTANTI PER LA PROGRAMMAZIONE BASIC AVANZATA: PROVIAMO A VEDERE COME SI ADOPERANO...

di G. RICCOBONO

I calcolatori MSX, al contrario di molti altri loro fratelli a otto BIT, offrono ai loro utenti una completa gamma di operatori logici (molto importanti nella programmazione BASIC avanzata). Fra questi possiamo trovare i fondamentalissimi AND, OR e NOT affiancati dai meno importanti ma egualmente validi XOR, EQV e IMP. Notiamo, per chi non avesse confidenza con questo argomento, che i secondi sono ottenibili da opportune combinazioni dei primi.

L'uso più comune, dei primi tre operatori logici, consiste nell'aggregare in maniera organizzata espressioni contenenti operatori relazionali; ricordate a tal fine che una qualsiasi espressione contenente operatori logici e relazionali darà come risultato una affermazione VERO oppure FALSO (rispettivamente rappresentate, come ricorderete, dai valori —1 e 0). Un secondo uso, comune stavolta a tutti e sei, consiste invece nel processare direttamente dei bit; ricordiamo in proposito che tali operazioni vanno svolte su numeri interi, ogni argomento che non sia un intero essendo convertito nel proprio complemento a due intero e di 16 bit (questi aspetti verranno comunque chiariti più avanti).

Cominciamo dunque con l'analizzare l'uso degli operatori NOT, AND e OR utilizzati, insieme agli operatori relazionali, nella costruzione di espressioni logiche.

L'USO DEGLI OPERATORI LOGICI

La maniera più semplice, per presentare questo tipo di utilizzo degli operatori logici, è di portare qualche esempio. L'istruzione:

```
PRINT NOT(1=1)
```

darà come risultato il valore 0, cioè FALSO, in quanto l'operatore NOT non fà altro che trasformare in FALSO l'espressione VERO e viceversa. Così, analogamente, l'istruzione:

```
PRINT NOT A
```

restituirà un valore VERO nel caso che A sia uguale a zero e FALSO in caso contrario.

Per quanto riguarda l'operatore AND possiamo dire che esso serve a "unire" due espressioni relazionali fra loro indipendenti e restituisce un valore VERO se e solo se entrambe le espressioni sono VERE. Se anche solo

una delle due è FALSA il valore restituito sarà FALSO. Così, ad esempio, l'istruzione:

X=10:Y=20:PRINT
(X=5) AND (Y=20)

darà come risultato FALSO. Per ottenere VERO dovremo scrivere:

PRINT (X=10) AND (Y=20).

L'operatore OR ha in un certo senso un ruolo complementare a quello dell'operatore AND, infatti anche l'operatore OR serve a "unire" due espressioni relazionali, ma fornisce un valore FALSO se e solo se entrambe le espressioni sono FALSE. In altre parole basta che una sola delle due espressioni sia VERA (o tutte e due) per ottenere come risultato VERO. Così, l'espressione:

X=10:Y=20:PRINT
(X=5) OR (Y=20)

darà come risultato il valore (—1), cioè VERO. Per ottenere come risultato FALSO, avremmo dovuto ad esempio scrivere:

PRINT (X=5) OR (Y=30).

I tre operatori appena presentati sono molto utilizzati, nello svolgimento dei programmi, abbinati a delle istruzioni IF... THEN... ELSE, in modo da poter effettuare decisioni e controllare il flusso logico del programma quando diverse condizioni devono combaciare.

OPERAZIONI LOGICHE SU BYTES

Un secondo modo di utilizzare gli operatori logici è quello di operare sui bit di un byte. Per meglio capire la cosa conviene vedere, prima di tutto, come i vari operatori logici agiscono su un singolo bit o su una coppia di singoli bit.

Cominciamo anche qui con l'operatore NOT. Come già visto, tale operatore cambia la proposizione VERO in FALSO e viceversa. Analogamente, quando utilizzata su un bit, ne cambia il valore da 0 a 1 e viceversa, a seconda del valore corrente. Chiamiamo ad esempio A un bit, se il bit A ha il valore 0 allora NOT A assumerà il valore 1 e viceversa, secondo la semplicissima tabellina:

A	NOT A
0	1
1	0

Analogamente, considerando una coppia di bit A e B avremo, per gli operatori AND e OR, le seguenti tabelline:

A	B	A AND B
0	0	0
0	1	0
1	0	0
1	1	1

A	B	A OR B
0	0	0
0	1	1
1	0	1
1	1	1

Come si vede, mettendo il valore 0 in relazione con la proposizione FALSO e il valore 1 in relazione con la proposizione VERO, vi è una forte analogia fra i due modi di agire dei tre operatori logici NOT, AND e OR.

Facendo un passo avanti vediamo a quale risultato portano gli operatori XOR, EQV e IMP. Anche questi tre, come OR e AND, operano su una coppia di bit e non sul bit singolo; guardate attentamente le tabelline seguenti per capire esattamente la loro funzione:

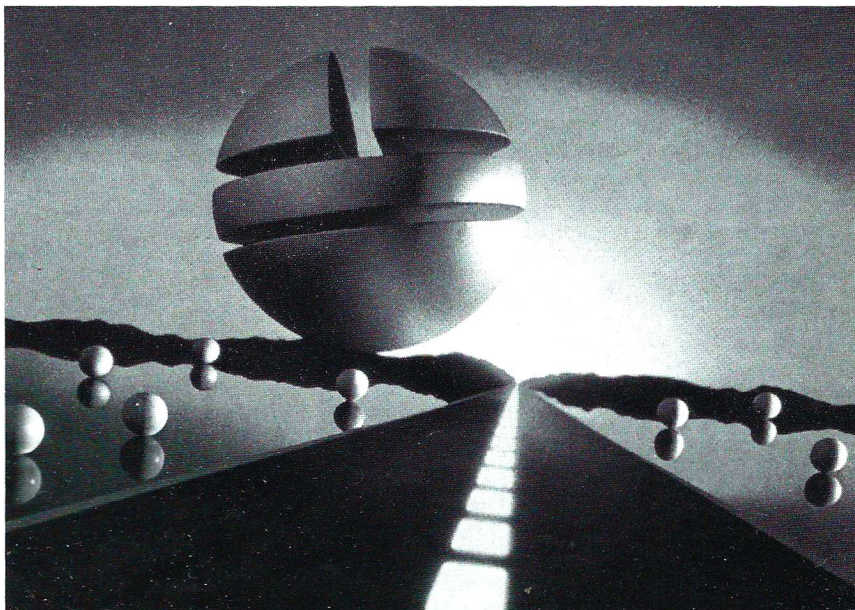
A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

A	B	A EQV B
0	0	1
0	1	0
1	0	0
1	1	1

A	B	A IMP B	B IMP A
0	0	1	1
0	1	1	0
1	0	0	1
1	1	1	1

Come vedete, nella tabellina dell'operatore IMP è stato riportato, oltre al risultato di A IMP B, anche il risultato di B IMP A, questo perché, a differenza di tutti gli altri operatori sin qui presentati, quest'ultimo non gode della proprietà commutativa.

ATTENTI AGLI ERRORI

Fatto troppo spesso sottovalutato, può portare in verità a gravi errori. Se infatti non siamo più che concentrati, possiamo essere portati a scrivere ad esempio A IMP B quando in verità vorremmo scrivere B IMP A. Questo può accadere a causa della nostra naturale tendenza a mettere la A prima della B, o la X prima della Y etc. Orbene, questo errore non porta nessuna conseguenza quando gli operatori in gioco sono OR, AND, XOR ed EQV (rifate le tabelline presentate sopra scambiando l'ordine di A e B per sincerarvene); fondamentalmente diverso si presenta invece il caso di IMP, che può portare al genere di errore menzionato quando A e B sono differenti l'uno dall'altro. ■

ASSEMBLER ERROR

Comprando i numeri della vostra rivista e leggendo il corso di linguaggio macchina mi è venuta voglia di impararlo. Ho comprato il programma assemblatore "CHAMP" (MSA) ed ho provato a copiare uno dei vostri listati (precisamente quello della rivista numero 9). Con mio grande disappunto non riesco a copiare tutte le righe perché, alcune, le segna errore (es.: errore READ: LD A,(DE) (eliminando i":" il listato ha proseguito) CALL 004DH errore (non sono riuscito ad eliminarlo)

Vorrei sapere il perché ed una eventuale soluzione.

Alessandro Accardo - Firenze

Gli errori riscontrati sul programma CHAMP sono dovuti ad incompatibilità di sintassi. Ovvero, il programma assemblatore utilizzato per scrivere i listati pubblicati su MSX Computer Magazine richiede i due punti (:) dopo il nome di una label, mentre il programma CHAMP no.

Allo stesso modo l'istruzione CALL 004DH per ZEN (è il nome dell'assemblatore da noi utilizzato) indica il salto alla subroutine allocata all'indirizzo esadecimale 004D. Dovrai quindi verificare qual è la sintassi di CHAMP per scrivere i numeri esadecimali.

ATTENZIONE ALL'OPEN

Utilizzo uno Spectravideo SVI 728 e non riesco ad usare la funzione FIELD. Infatti aprendo un file ad accesso diretto con l'istruzione:

*100 OPEN F$ AS#1**

il computer continua a darmi un messaggio di errore del tipo "BAD FILE NAME IN 100".

Francesco Celiberto
Cassano Jonio (CS)

L'errore "BAD FILE NAME IN 100" non è dato perché si è usata male l'istruzione FIELD ma per l'utilizzo scorretto dell'istruzione OPEN. Infatti la linea 100,

100 OPEN F$ AS#1

ammesso che F$ sia "A:nome del file" o "B:nome del file", è incompleta. La forma corretta è:

100 OPEN F$ AS#1 LEN=lrec

dove lrec è la lunghezza del record del file che si vuole aprire. Dopo di che va usata l'istruzione FIELD come segue:

110 FIELD 1#,lrec AS var1$ [,lrec 2 AS var 2$]...

dove lrec 1, [lrec 2]... sono porzioni del record, ovvero le lunghezze dei campi, a loro volta assegnate alle variabili di lettura var 1$ [,var 2$]...

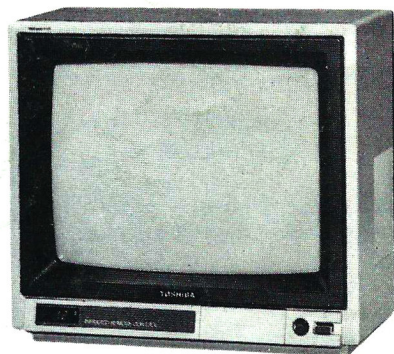
SPRITE & GIOCHI

Vorrei sapere come devo sviluppare un programma in modo che tra collisioni di diversi sprite si abbiano diverse conseguenze.

Esempio: avendo 3 sprite A, B, e C se A e B entrano in collisione termina il gioco;
se B e C entrano in collisione non accade nulla;
se C e A entrano in collisione esplode solo C.

Paolo Mattiazzi - Maerne (VE)

Quando due sprite entrano in collisione? Quando si toccano: in altre parole quando la loro distanza, espressa in pixel, è minore di 0. Supponiamo di avere tre sprite (A, B, C) di formato piccolo, 8x8 pixel, e di voler decidere quali sprite si sono "scontrati". Definendo alla linea 1000 l'inizio della subroutine di gestione collisione (chiamata con ON SPRITE GOSUB 1000) scriveremo:

1000 IF ABS(XA-XB)<8 THEN...
COLLISIONE TRA LO SPRITE A E B
1010 IF ABS(XB-XC)<8 THEN...
COLLISIONE TRA LO SPRITE B E C
1020 IF ABS(XC-XA)<8 THEN...
COLLISIONE TRA LO SPRITE C E A
1030 RETURN

Dove XA, XB e XC sono rispettivamente le coordinate sull'asse X dello sprite A, B e C. Il sottoprogramma calcola il valore assoluto (ABS) tra la distanza in X tra uno sprite e l'altro. Se tale distanza è minore di 8 pixel, allora i due sprite considerati sono entrati in collisione perché la distanza minima tra due sprite affinché non si sovrappongano, deve essere di 8 pixel. Dovendo usare gli sprite grandi (16x16 pixel) al posto del valore 8 (linee 1000, 1010, 1020), scriveremo 16.

HARDCOPY SU 80 COLONNE

Nel N. 4 di Computer Magazine ho trovato un programma per poter eseguire una copia dello schermo grafico su stampante. Purtroppo io possiedo una stampante a ottanta colonne e perciò il disegno mi viene stampato soltanto fino a metà foglio. Malgrado i miei tentativi, non sono riuscito ancora a far sì che il disegno mi venga stampato su tutto il foglio.

??? Cristian - Trieste

La routine di hardcopy pubblicata sul n. 4 di MSX Computer Magazine è stata sviluppata su una stampante ad ottanta colonne tenendo però conto di coloro che dispongono di una stampante a quaranta colonne. Il programma non può quindi essere modificato. L'unica soluzione è cercare di individuare la parte che ricava i valori binari da stampare e raddoppiare tale risultato inviandolo, successivamente, alla periferica.